

NNN		NNN	MMM	MMM	LLL
NNN		NNN	MMM	MMM	LLL
NNN		NNN	MMM	MMM	LLL
NNN		NNN	MMMMMM	MMMMMM	LLL
NNN		NNN	MMMMMM	MMMMMM	LLL
NNN		NNN	MMMMMM	MMMMMM	LLL
NNNNNN		NNN	MMM	MMM	LLL
NNNNNN		NNN	MMM	MMM	LLL
NNNNNN		NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLL
NNN	NNNNNN	NNN	MMM	MMM	LLL
NNN	NNNNNN	NNN	MMM	MMM	LLL
NNN	NNNNNN	NNN	MMM	MMM	LLL
NNN	NNNNNN	NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLLLLLLLLLLLLLLL
NNN	NNN	NNN	MMM	MMM	LLLLLLLLLLLLLLLL
NNN	NNN	NNN	MMM	MMM	LLLLLLLLLLLLLLLL

_S

Ps

NP

NP

\$G

\$O

NP

PA

_L

NMI
VO4

: 1

```

LL               IIIIII               SSSSSSSS
LL               IIIIII               SSSSSSSS
LL               II                    SS
LL               II                    SS
LL               II                    SS
LL               II                    SS
LL               II                    SSSSSS
LL               II                    SSSSSS
LL               II                    SS
LL               II                    SS
LL               II                    SS
LL               II                    SS
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS

```



```
0001 0 %TITLE 'Network Management Listener module to forward NICE messages'
0002 0 MODULE NML$FORWARD (
0003 0     ADDRESSING_MODE (NONEXTERNAL=GENERAL),
0004 0     ADDRESSING_MODE (EXTERNAL=GENERAL),
0005 0     IDENT = 'V04-000') =
0006 0
0007 1 BEGIN
0008 1
0009 1 |
0010 1 |*****
0011 1 |*
0012 1 |*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0013 1 |*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0014 1 |*  ALL RIGHTS RESERVED.
0015 1 |*
0016 1 |*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0017 1 |*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0018 1 |*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0019 1 |*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0020 1 |*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0021 1 |*  TRANSFERRED.
0022 1 |*
0023 1 |*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0024 1 |*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0025 1 |*  CORPORATION.
0026 1 |*
0027 1 |*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0028 1 |*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0029 1 |*
0030 1 |*
0031 1 |*****
0032 1 |
0033 1 |
0034 1 |++
0035 1 |FACILITY:  DECnet-VAX Network Management Listener
0036 1 |
0037 1 |ABSTRACT:
0038 1 |
0039 1 |    This module forwards NICE messages from NCP to two other programs.
0040 1 |    These programs are:
0041 1 |
0042 1 |    The Maintenance Operations Module (MOM):
0043 1 |    It's function is to perform maintenance functions such as down line
0044 1 |    load, up line dump, trigger, and loop line, circuit or node.
0045 1 |
0046 1 |    The NI Configurator Module:
0047 1 |    It's function is to gather information about the various circuits
0048 1 |    on the NI and, when requested, return this information to NCP. NML
0049 1 |    is a conduit for the request and the returned information. In this
0050 1 |    module, NML establishes a logical link to the NI Configurator Module
0051 1 |    and forwards the NICE message from NCP to it. It then takes whatever
0052 1 |    responses returned by the NI Configurator Module, and sends them back
0053 1 |    to NCP.
0054 1 |
0055 1 |ENVIRONMENT:  VAX/VMS Operating System
0056 1 |
0057 1 |AUTHOR:  Kathy Perko
```

```
.. 58 0058 1 |
.. 59 0059 1 | CREATION DATE: 17-Jan-1983
.. 60 0060 1 |
.. 61 0061 1 | MODIFIED BY:
.. 62 0062 1 | V03-006 MKP0006 Kathy Perko 11-April-1984
.. 63 0063 1 | Add NCP version to buffer passed to MOM.
.. 64 0064 1 |
.. 65 0065 1 | V03-005 MKP0005 Kathy Perko 4-Mar-1984
.. 66 0066 1 | Don't create a new mailbox every time MOM is invoked. It
.. 67 0067 1 | eats up bytln quota.
.. 68 0068 1 |
.. 69 0069 1 | V03-004 MKP0004 Kathy Perko 3-Jan-1984
.. 70 0070 1 | Convert old $TRNLOG system service to new $TRNLNM for
.. 71 0071 1 | translating logical names.
.. 72 0072 1 |
.. 73 0073 1 | V03-003 MKP0003 Kathy Perko 10-May-1983
.. 74 0074 1 | Fix mailbox communication with MOM so that NML$MOM_MBX
.. 75 0075 1 | goes into the process logical name table instead of the
.. 76 0076 1 | group logical name table. This will force multiple incarnations
.. 77 0077 1 | of NML and MOM to use separate mailboxes.
.. 78 0078 1 |
.. 79 0079 1 | V03-002 MKP0002 Kathy Perko 29-April-1983
.. 80 0080 1 | Upcase the logical name of the mailbox used to communicate
.. 81 0081 1 | with MOM.
.. 82 0082 1 |
.. 83 0083 1 | V03-001 MKP0001 Kathy Perko 21-April-1983
.. 84 0084 1 | Use a command procedure when invoking MOM. Also, if
.. 85 0085 1 | the SPAWN to start up MOM fails, assume it's being run
.. 86 0086 1 | from a batch job, and start it up again with null input
.. 87 0087 1 | and output devices.
.. 88 0088 1 |
.. 89 0089 1 | --
```



```

91      0090 1 %SBTTL 'Declarations'
92      0091 1
93      0092 1
94      0093 1  INCLUDE FILES:
95      0094 1
96      0095 1
97      0096 1  LIBRARY 'LIB$:NMLLIB';
98      0097 1  LIBRARY 'SHRLIB$:NMLIBRY';
99      0098 1  LIBRARY 'SHRLIB$:NET';
100     0099 1  LIBRARY 'SYSS$LIBRARY:STARLET';
101     0100 1
102     0101 1
103     0102 1  TABLE OF CONTENTS:
104     0103 1
105     0104 1
106     0105 1  FORWARD ROUTINE
107     0106 1      nml$call_mom: NOVALUE,
108     0107 1      nml$call_ni_config: NOVALUE,
109     0108 1      nml_open_config_link: NOVALUE,
110     0109 1      nml_config_qio,
111     0110 1      nml_chkerr: NOVALUE;
112     0111 1
113     0112 1  Externals
114     0113 1
115     0114 1  $nml_extdef;
116     0115 1
117     0116 1
118     0117 1  EXTERNAL
119     0118 1      nml$gb_ncp_version;
120     0119 1
121     0120 1  EXTERNAL LITERAL
122     0121 1      nml$_opabterm;
123     0122 1
124     0123 1  EXTERNAL ROUTINE
125     0124 1      LIB$SPAWN,
126     0125 1      LIB$ASN_WITH_MBX,
127     0126 1      nml$bld_reply,
128     0127 1      nml$send,
129     0128 1      nml$debug_msg;
130     0129 1
131     0130 1  LITERAL
132     0131 1      nml$_maxmbxmsg = 200;
133     0132 1
134     0133 1  OWN
135     0134 1      nml$_mom_mbx_chan: WORD INITIAL (0), ! Channel to Mailbox for communicating
136     0135 1                                     with MOM.
137     0136 1      nml$_config_chan: WORD, ! Logical link channel to NICONFIG.
138     0137 1      nml$_mbx_chan: WORD, ! Logical link's Mailbox channel.
139     0138 1      nml$_q_mbx_iosb: $iosb, ! IOSB for mailbox QIOs.
140     0139 1      nml$_a_mbxmsg: ! Mailbox message buffer.
141     0140 1      VECTOR [nml$_maxmbxmsg, BYTE];
142     0141 1
143     0142 1
144     0143 1  MACRO
145     0144 1      $nml_niconfig_ncb =
146     0145 1      %STRING ('::',
147     0146 1      'TASK=$NICONFIG/',

```

NML\$FORWARD
V04-000

Network Management Listener module to forward N
Declarations

F 13
16-Sep-1984 00:15:46
14-Sep-1984 12:50:09

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLFORWRD.B32;1

Page 4
(2)

```
: 148      M 0147 1      %CHAR (0,0),      ! Word of zero
: 149      M 0148 1      )
: 150      M 0149 1      %;
: 151      M 0150 1
: 152      M 0151 1 BIND
: 153      M 0152 1      nml$q_ncb = UPLIT (
: 154      M 0153 1      LONG (%CHARCOUNT ($nml_niconfig_ncb),
: 155      M 0154 1      UPLIT PSECT ($OWNS) ($nml_niconfig_ncb))
: 156      M 0155 1      );
: 157      M 0156 1
```



```
159 0157 1 %SBTTL 'NML$CALL_MOM Routine to invoke Maintenance Operations Module'
160 0158 1 GLOBAL ROUTINE NML$CALL_MOM: NOVALUE =
161 0159 1
162 0160 2 BEGIN
163 0161 2
164 0162 2 !++
165 0163 2 FUNCTIONAL DESCRIPTION:
166 0164 2 The Maintenance Operations Module (MOM) is a separate program from
167 0165 2 NML and NCP. It's function is to perform various maintance operations
168 0166 2 such as down line load, up line dump, trigger, and loop circuit, node,
169 0167 2 or line. For operator requested maintenance functions, NML is
170 0168 2 a conduit for the NICE request and response. In this module, NML
171 0169 2 establishes a mailbox to which it writes the NICE messages, and
172 0170 2 then spawns MOM. MOM performs the function and puts a NICE response
173 0171 2 in the mailbox. NML then forwards this NICE response to NCP.
174 0172 2
175 0173 2 FORMAL PARAMETERS:
176 0174 2 None
177 0175 2
178 0176 2 IMPLICIT INPUTS:
179 0177 2 The NICE message in nml$ab_rcvbuffer.
180 0178 2
181 0179 2 IMPLICIT OUTPUTS:
182 0180 2 A NICE message is sent to NCP.
183 0181 2
184 0182 2 SIDE EFFECTS:
185 0183 2 The Maintenance Operations Module (MOM) is run.
186 0184 2
187 0185 2 --
188 0186 2
189 0187 2 FIELD
190 0188 2 itmlst_fields =
191 0189 2 SET
192 0190 2 itm_buf_len = [0,0,16,0],
193 0191 2 itm_item_code = [2,0,16,0],
194 0192 2 itm_buf_add = [4,0,32,0],
195 0193 2 itm_ret_len = [8,0,32,0],
196 0194 2 itm_list_end = [12,0,32,0]
197 0195 2 TES;
198 0196 2
199 0197 2 LOCAL
200 0198 2 status,
201 0199 2 mom_status,
202 0200 2 msg_len,
203 0201 2 getdvi_itmlst: BBLOCK [4] FIELD (itmlst_fields),
204 0202 2 crelnm_itmlst: BBLOCK [4] FIELD (itmlst_fields);
205 0203 2
206 0204 2 OWN
207 0205 2 mbx_name: BBLOCK [64];
208 0206 2
209 0207 2
210 0208 2 Create mailbox with which to communicate with MOM. Create a logical name
211 0209 2 for the mailbox in the process logical name table. MOM inherits NML's logical
212 0210 2 names as a result of the SPAWN, and putting the logical name in the process
213 0211 2 table makes sure that other incarnations of NML and MOM do not use this
214 0212 2 mailbox.
215 0213 2
```

```
216 0214 2 IF .nml$w_mom_mbx_chan EQL 0 THEN
217 0215 BEGIN
218 P 0216 status = $CREMBX (CHAN = nml$w_mom_mbx_chan,
219 P 0217 MAXMSG = nml$k_rcvbuflen, ! Max length for a NICE message.
220 0218 PROMSK = %B'1111111100000000'); ! Protection = S:RWED, O:RWED, G, W
221 0219 nml_chkerr (.status, 0);
222 0220 END;
223 0221 getdvi_itmlst [itm_buf_len] = 64;
224 0222 getdvi_itmlst [itm_item_code] = dvi$devnam;
225 0223 getdvi_itmlst [itm_buf_add] = mbx_name;
226 0224 getdvi_itmlst [itm_ret_len] = crelnm_itmlst [itm_buf_len];
227 0225 getdvi_itmlst [itm_list_end] = 0;
228 P 0226 status = $GETDVI (CHAN = .nml$w_mom_mbx_chan,
229 0227 ITMLST = getdvi_itmlst);
230 0228 nml_chkerr (.status, 0);
231 0229 crelnm_itmlst [itm_item_code] = lnm$string;
232 0230 crelnm_itmlst [itm_buf_add] = mbx_name;
233 0231 crelnm_itmlst [itm_ret_len] = crelnm_itmlst [itm_buf_len];
234 0232 crelnm_itmlst [itm_list_end] = 0;
235 P 0233 status = $CRELNM (TABNAM = %ASCII 'LNMS$PROCESS_TABLE', ! Process logical name table,
236 P 0234 LOGNAM = %ASCII 'NML$MOM_MBX',
237 0235 ITMLST = crelnm_itmlst);
238 0236 nml_chkerr (.status, 0);
239 0237
240 0238 ! Put the NCP network management version number at the beginning of the NICE
241 0239 ! message being passed to MOM.
242 0240
243 0241 CH$MOVE (.nml$gl_rcvdatlen, nml$ab_rcvbuffer, nml$ab_rcvbuffer+3);
244 0242 CH$MOVE (3, nml$gb_ncp_version, nml$ab_rcvbuffer);
245 0243 msg_len = .nml$gl_rcvdatlen + 3;
246 0244
247 0245 ! Write NICE message to mailbox
248 0246
249 0247 nml_config_qio (.nml$w_mom_mbx_chan,
250 0248 io$writevblk OR io$m_now,
251 0249 nml$ab_rcvbuffer,
252 0250 msg_len);
253 0251
254 0252 ! Spawn the Maintenance Operations Module (MOM). MOM will translate the
255 0253 ! logical name, NML$MOM_MBX, and then read the NICE message and process it.
256 0254 ! When it is done, it will write a response NICE message to the mailbox.
257 0255
258 0256 status = LIB$SPAWN (%ASCII '$ @SYSS$SYSTEM:MOM.COM',
259 0257 0,0,0,0,0,
260 0258 mom_status);
261 0259 IF NOT .status THEN
262 0260 status = LIB$SPAWN (%ASCII '$ @SYSS$SYSTEM:MOM.COM',
263 0261 %ASCII 'NL:', ! Null input device
264 0262 %ASCII 'NL:', ! Null output device
265 0263 0,0,0,
266 0264 mom_status);
267 0265 nml_chkerr (.status, mom_status);
268 0266
269 0267 ! Read mailbox to get the NICE response MOM puts there when it's finished.
270 0268
271 0269 msg_len = nml$k_sndbflen;
272 0270 nml_config_qio T.nml$w_mom_mbx_chan,
```



```

: 273      0271      2      io$ readvblk OR io$m_now,
: 274      0272      2      nml$ab_sndbuffer,
: 275      0273      2      msg_len);
: 276      0274      2      |
: 277      0275      2      | Check to make sure that the message I got back isn't the one I just
: 278      0276      2      | wrote to the mailbox. This can happen if MOM isn't successfully
: 279      0277      2      | started up.
: 280      0278      2      |
: 281      0279      2      | IF CH$EQL (.nml$gl_rcvdatlen, nml$ab_rcvbuffer,
: 282      0280      2      | .msg_len, nml$ab_sndbuffer, 0) THEN
: 283      0281      2      |     nml_chkerr (ss$_endoffile, 0);
: 284      0282      2      |
: 285      0283      2      | Send msg to NCP.
: 286      0284      2      |
: 287      0285      2      | nml$send (nml$ab_sndbuffer, .msg_len);
: 288      0286      1      | END;
                        | of nml$call_mom
```

```

;
.TITLE NML$FORWARD Network Management Listener module
      to forward N
.IDENT \V04-000\
.PSECT $PLITS,NOWRT,NOEXE,2
      00000015 00000 P.AAA: .LONG 21
      00000000' 00004 .ADDRESS P.AAB
42 41 54 5F 53 53 45 43 4F 52 50 24 4D 4E 4C 00008 P.AAD: .ASCII \LNMS$PROCESS_TABLE\<0><0><0>
      00 00 00 45 4C 00017
      010E0011 0001C P.AAC: .LONG 17694737
      00000000' 00020 .ADDRESS P.AAD
      00 58 42 4D 5F 4D 4F 4D 24 4C 4D 4E 00024 P.AAF: .ASCII \NML$MOM_MBX\<0>
      010E000B 00030 P.AAE: .LONG 17694731
      00000000' 00034 .ADDRESS P.AAF
4D 3A 4D 45 54 53 59 53 24 53 59 53 40 20 24 00038 P.AAH: .ASCII \S @SYSS$SYSTEM:MOM.COM\<0><0><0>
      00 00 00 4D 4F 43 2E 4D 4F 00047
      010E0015 00050 P.AAG: .LONG 17694741
      00000000' 00054 .ADDRESS P.AAH
4D 3A 4D 45 54 53 59 53 24 53 59 53 40 20 24 00058 P.AAJ: .ASCII \S @SYSS$SYSTEM:MOM.COM\<0><0><0>
      00 00 00 4D 4F 43 2E 4D 4F 00067
      010E0015 00070 P.AAI: .LONG 17694741
      00000000' 00074 .ADDRESS P.AAJ
      00 3A 4C 4E 00078 P.AAL: .ASCII \NL:\<0>
      010E0003 0007C P.AAK: .LONG 17694723
      00000000' 00080 .ADDRESS P.AAL
      00 3A 4C 4E 00084 P.AAN: .ASCII \NL:\<0>
      010E0003 00088 P.AAM: .LONG 17694723
      00000000' 0008C .ADDRESS P.AAN
.PSECT $OWNS,NOEXE,2
      0000 00000 NML$W_MOM_MBX_CHAN:
      .WORD 0
      00002 NML$W_CONFIG_CHAN:
      .BLKB 2
      00004 NML$W_MBXCHAN:
      .BLKB 2
      00006 .BLKB 2
```

NML\$FORWARD
V04-000

Network Management Listener module to forward N 16-Sep-1984 00:15:46
NML\$CALL_MOM Routine to invoke Maintenance Op 14-Sep-1984 12:50:09

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLFORWRD.B32;1

Page 8
(3)

```

                                J 13
00008 NML$Q_MBX_IOSB:          8
                                .BLKB
00010 NML$A_MBXMSG:           200
                                .BLKB
46  4E  4F  43  49  4E  24  3D  4B  53  41  54  22  3A  3A  000D8 P.AAB: .ASCII \::'TASK=$NICONFIG/\<0><0>\''\<0><0>
   00  00  22  00  00  2F  47  49  000E7
                                000EF .ASCII <0>
                                000F0 MBX_NAME: .BLKB 64
                                P.AAA
                                NML$Q_NCB=
                                .EXTRN NML$GB_EVTSRCTYP
                                .EXTRN NML$GQ_EVTSRCDS
                                .EXTRN NML$GW_EVTCLASS
                                .EXTRN NML$GB_EVTMSKTYP
                                .EXTRN NML$GQ_EVTMSKDSC
                                .EXTRN NML$GW_EVTSNKADR
                                .EXTRN NML$GW_ACP_CHAN
                                .EXTRN NML$GL_LOGMASK, NML$GQ_ENTSTRDSC
                                .EXTRN NML$AB_QIOBUFFER
                                .EXTRN NML$GQ_QIOBFDSC
                                .EXTRN NML$AB_EXEBUFFER
                                .EXTRN NML$GL_EXEDATPTR
                                .EXTRN NML$GQ_EXEDATDSC
                                .EXTRN NML$GQ_EXEBFDSC
                                .EXTRN NML$AB_RCVBUFFER
                                .EXTRN NML$GQ_RCVBFDSC
                                .EXTRN NML$AB_SNDBUFFER
                                .EXTRN NML$GQ_SNDBFDSC
                                .EXTRN NML$GL_RCVDATLEN
                                .EXTRN NML$AB_CPTABLE, NML$AB_MSGBLOCK
                                .EXTRN NML$AB_ENTITY_ID
                                .EXTRN NML$AB_QUALIFIER_ID
                                .EXTRN NML$AB_ENTITYDATA
                                .EXTRN NML$AB_NML_NMV, NML$AB_PRMSEM
                                .EXTRN NML$AB_RECBUF, NML$AL_ENTINFNTAB
                                .EXTRN NML$AL_PERMINFTAB
                                .EXTRN NML$AW_PRM_DES, NML$GB_CMD_VER
                                .EXTRN NML$GB_ENTITY_CODE
                                .EXTRN NML$GB_ENTITY_FORMAT
                                .EXTRN NML$GL_QUALIFIER_PST
                                .EXTRN NML$GB_QUALIFIER_FORMAT
                                .EXTRN NML$GB_FUNCTION
                                .EXTRN NML$GB_INFO, NML$GB_OPTIONS
                                .EXTRN NML$GL_PRMCODE, NML$GL_PRS_FLGS
                                .EXTRN NML$GL_NML_ENTITY
                                .EXTRN NML$GQ_NETNAMDSC
                                .EXTRN NML$GQ_RECBFDSC
                                .EXTRN NML$GW_PRMDESCNT
                                .EXTRN NML$GB_NCP_VERSION
                                .EXTRN NML$ OPABTERM, LIB$SPAWN
                                .EXTRN LIB$ASN_WTH_MBX
                                .EXTRN NML$BLD_REPCY, NML$SEND
                                .EXTRN NML$DEB0G_MSG, SYSSCREMBX
                                .EXTRN SYSSGETDVT, SYSSCRELNM
                                .PSECT $CODE$,NOWRT,2
```


				OFFC 00000	.ENTRY	NML\$CALL_MOM, Save R2,R3,R4,R5,R6,R7,R8,R9,-;		
				5B 00000000G	00 9E 00002	MOVAB	R10,R11	0158
				5A 00000000V	00 9E 00009	MOVAB	NML\$AB_RCVBUFFER, R11	
				59 00000000'	00 9E 00010	MOVAB	NML_CHKERR, R10	
				58 00000000'	00 9E 00017	MOVAB	P.AAE, R9	
				5E	10 C2 0001E	MOVAB	NML\$W_MOM_MBX_CHAN, R8	
					68 B5 00021	SUBL2	#16, SP	
					23 12 00023	TSTW	NML\$W_MOM_MBX_CHAN	0214
					7E 7C 00025	BNEQ	1\$	
				7E FF00	8F 3C 00027	CLRQ	-(SP)	0218
					7E D4 0002C	MOVZWL	#65280, -(SP)	
				7E 0200	8F 3C 0002E	CLRL	-(SP)	
					58 DD 00033	MOVZWL	#512, -(SP)	
					7E D4 00035	PUSHL	R8	
				00000000G	00	CLRL	-(SP)	
				57	07 FB 00037	CALLS	#7, SYSSCREMBX	
					50 D0 0003E	MOVL	R0, STATUS	
					7E D4 00041	CLRL	-(SP)	0219
					57 DD 00043	PUSHL	STATUS	
				6A	02 FB 00045	CALLS	#2, NML_CHKERR	
				6E 00200040	8F D0 00048	MOVL	#2097216, GETDVI_ITMLST	0221
				04 AE 00F0	C8 9E 0004F	MOVAB	MBX_NAME, GETDVI_ITMLST+4	0223
				08 AE 04	AE 9E 00055	MOVAB	CRELNM_ITMLST, GETDVI_ITMLST+8	0224
					OC AE D4 0005A	CLRL	GETDVI_ITMLST+12	0225
					7E 7C 0005D	CLRQ	-(SP)	0227
					7E 7C 0005F	CLRQ	-(SP)	
				10	AE 9F 00061	PUSHAB	GETDVI_ITMLST	
					7E D4 00064	CLRL	-(SP)	
				7E	68 3C 00066	MOVZWL	NML\$W_MOM_MBX_CHAN, -(SP)	
					7E D4 00069	CLRL	-(SP)	
				00000000G	00	CALLS	#8, SYSSGETDVI	
				57	08 FB 0006B	MOVL	R0, STATUS	
					50 D0 00072	CLRL	-(SP)	0228
					7E D4 00075	PUSHL	STATUS	
					57 DD 00077	CALLS	#2, NML_CHKERR	
				6A	02 FB 00079	MOVW	#2, CRELNM_ITMLST+2	0229
				06 AE 00F0	C8 9E 0007C	MOVAB	MBX_NAME, CRELNM_ITMLST+4	0230
				08 AE 04	AE 9E 00080	MOVAB	CRELNM_ITMLST, CRELNM_ITMLST+8	0231
				OC AE 04	6D D4 0008B	CLRL	CRELNM_ITMLST+12	0232
					AE 9F 0008D	PUSHAB	CRELNM_ITMLST	0235
					7E D4 00090	CLRL	-(SP)	
					59 DD 00092	PUSHL	R9	
				EC	A9 9F 00094	PUSHAB	P.AAC	
					7E D4 00097	CLRL	-(SP)	
				00000000G	00	CALLS	#5, SYSSCRELNM	
				57	05 FB 00099	MOVL	R0, STATUS	
					50 D0 000A0	CLRL	-(SP)	0236
					7E D4 000A3	PUSHL	STATUS	
					57 DD 000A5	CALLS	#2, NML_CHKERR	
				6A	02 FB 000A7	MOVL	NML\$GL_RCVDATLEN, R6	0241
				56 00000000G	00 D0 000AA	MOVC3	R6, NML\$AB_RCVBUFFER, NML\$AB_RCVBUFFER+3	
				6B	56 28 000B1	INSV	NML\$GB_NCP-VERSION, #0, #24, -	0242
				00 00000000G	00 F0 000B6		NML\$AB_RCVBUFFER	
						MOVAB	3(R6), MSG_LEN	0243
				OC AE 03	A6 9E 000BF	PUSHAB	MSG_LEN	0247
					OC AE 9F 000C4	PUSHL	R11	
					5B DD 000C7			

NML\$FORWARD
V04-000

Network Management Listener module to forward N 16-Sep-1984 00:15:46
NML\$CALL_MOM Routine to invoke Maintenance Op 14-Sep-1984 12:50:09

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLFORWRD.B32;1

Page 10
(3)

	7E	70	8F	9A	000C9	MOVZBL	#112, -(SP)	0248
	7E		68	3C	000CD	MOVZWL	NML\$W MOM MBX_CHAN, -(SP)	0247
00000000V	00		04	FB	000D0	CALLS	#4, NML_CONFIG_QIO	
		08	AE	9F	000D7	PUSHAB	MOM STATUS	0256
			7E	7C	000DA	CLRQ	-(SP)	
			7E	7C	000DC	CLRQ	-(SP)	
			7E	D4	000DE	CLRL	-(SP)	
00000000G	00	20	A9	9F	000E0	PUSHAB	P.AAG	
	57		07	FB	000E3	CALLS	#7, LIB\$SPAWN	
	1A		50	D0	000EA	MOVL	R0, STATUS	0259
		08	57	E8	000ED	BLBS	STATUS, 2\$	0260
			AE	9F	000F0	PUSHAB	MOM STATUS	
			7E	7C	000F3	CLRQ	-(SP)	
			7E	D4	000F5	CLRL	-(SP)	
		58	A9	9F	000F7	PUSHAB	P.AAM	0261
		4C	A9	9F	000FA	PUSHAB	P.AAK	0260
		40	A9	9F	000FD	PUSHAB	P.AAI	
00000000G	00		07	FB	00100	CALLS	#7, LIB\$SPAWN	
	57		50	D0	00107	MOVL	R0, STATUS	
		08	AE	9F	0010A	PUSHAB	MOM STATUS	0265
			57	DD	0010D	PUSHL	STATUS	
	6A		02	FB	0010F	CALLS	#2, NML_CHKERR	
OC	AE	0200	8F	3C	00112	MOVZWL	#512, MSG_LEN	0269
		OC	AE	9F	00118	PUSHAB	MSG_LEN	0270
		00000000G	00	9F	0011B	PUSHAB	NML\$AB_SNDBUFFER	
	7E	71	8F	9A	00121	MOVZBL	#113, -(SP)	0271
	7E		68	3C	00125	MOVZWL	NML\$W MOM MBX_CHAN, -(SP)	0270
OC	AE	00	04	FB	00128	CALLS	#4, NML_CONFIG_QIO	
		00	00	2D	0012F	CMPC5	NML\$GL RCVDATLEN, NML\$AB_RCVBUFFER, #0, -	0279
		00000000G	00		00139		MSG_LEN, NML\$AB_SNDBUFFER	
		00000000G	00		0013E	BNEQ	3\$	
			0A	12	0013E	CLRL	-(SP)	0281
	7E	0870	7E	D4	00140	MOVZWL	#2160, -(SP)	
	6A		8F	3C	00142	CALLS	#2, NML_CHKERR	
		OC	02	FB	00147	PUSHL	MSG_LEN	0285
		00000000G	AE	DD	0014A	PUSHAB	NML\$AB_SNDBUFFER	
00000000G	00		00	9F	0014D	CALLS	#2, NML\$SEND	
			02	FB	00153	RET		0286
			04	00	0015A			

; Routine Size: 347 bytes, Routine Base: \$CODE\$ + 0000


```
0287 1 %SBTTL 'NML$CALL_NI_CONFIG Routine to talk to NI Configurator Module'
0288 1 GLOBAL ROUTINE NML$CALL_NI_CONFIG: NOVALUE =
0289 1
0290 2 BEGIN
0291 2
0292 2 !++
0293 2 FUNCTIONAL DESCRIPTION:
0294 2     This routine is called when NML receives a
0295 2     SET/SHOW MODULE CONFIGURATOR command.
0296 2     It establishes a logical link to the NI Configurator Module
0297 2     (NICONFIG), and then drives the process of sending and receiving
0298 2     NICE messages between NCP and NICONFIG.
0299 2
0300 2 FORMAL PARAMETERS:
0301 2     NONE
0302 2
0303 2 IMPLICIT INPUTS:
0304 2     The NICE message in nml$ab_rcvbuffer.
0305 2
0306 2 IMPLICIT OUTPUTS:
0307 2     NICE response message(s) from NICONFIG in nml$ab_sndbuffer.
0308 2
0309 2 ROUTINE VALUE:
0310 2 COMPLETION CODES:
0311 2
0312 2     NONE
0313 2
0314 2 SIDE EFFECTS:
0315 2
0316 2     NONE
0317 2
0318 2 --
0319 2
0320 2 LOCAL
0321 2     msg_len;
0322 2
0323 2
0324 2 Open a logical link to configurator module.
0325 2
0326 2 nml_open_config_link ();
0327 2
0328 2 Send the NICE message to the NI Configurator Module via the logical
0329 2 link just established.
0330 2
0331 2 nml_config_qio (.nml$w_config_chan,
0332 2               io$writevblk,
0333 2               nml$ab_rcvbuffer,
0334 2               nml$gl_rcvdatlen);
0335 2
0336 2
0337 2 Now read the response message (or messages) returned by the
0338 2 NICONFIG, and forward them to NCP.
0339 2
0340 2 msg_len = nml$sk_sndbflen;
0341 2 nml_config_qio (.nml$w_config_chan,
0342 2               io$readvblk,
0343 2               nml$ab_sndbuffer,
0344 2               msg_len);
0345 2
0346 2
```

```

: 347 0344 2 |
: 348 0345 2 | If NICONFIG is returning multiple responses, go into a loop until all
: 349 0346 2 | have been forwarded to NCP. Note that the "more" and "done" messages
: 350 0347 2 | are not forwarded because NML already sends them on it's own.
: 351 0348 2 |
: 352 0349 2 | IF .nml$ab_sndbuffer <0,8> EQL nma$c_sts_mor THEN
: 353 0350 3 | BEGIN
: 354 0351 3 | WHILE true DO
: 355 0352 4 | BEGIN
: 356 0353 4 | msg_len = nml$k_sndbflen;
: 357 0354 4 | nml_config_qio ? .nml$w_config_chan,
: 358 0355 4 | io$readvblk,
: 359 0356 4 | nml$ab_sndbuffer,
: 360 0357 4 | msg_len);
: 361 0358 4 |
: 362 0359 4 | | When NICONFIG returns a "done" message, exit. A "done" message
: 363 0360 4 | | is sent to NCP later.
: 364 0361 4 |
: 365 0362 4 | IF .nml$ab_sndbuffer <0,8> EQL (nma$c_sts_don AND %X'FF') THEN
: 366 0363 4 | EXITLOOP
: 367 0364 4 | ELSE
: 368 0365 4 | | Forward NICONFIG's response to NCP.
: 369 0366 4 | |
: 370 0367 4 | | nml$send (nml$ab_sndbuffer, .msg_len);
: 371 0368 4 | |
: 372 0369 3 | END;
: 373 0370 3 | END
: 374 0371 2 | ELSE
: 375 0372 2 | | Send msg to NCP.
: 376 0373 2 | |
: 377 0374 2 | | nml$send (nml$ab_sndbuffer, .msg_len);
: 378 0375 2 | |
: 379 0376 1 | END;
: | of nml$call_ni_config
```

		003C 00000	.ENTRY NML\$CALL_NI_CONFIG, Save R2,R3,R4,R5	: 0288
55	00000000G	00 9E 00002	MOVAB NML\$SEND, R5	:
54	00000000V	00 9E 00009	MOVAB NML_CONFIG_QIO, R4	:
53	00000000'	00 9E 00010	MOVAB NML\$W_CONFIG_CHAN, R3	:
52	00000000G	00 9E 00017	MOVAB NML\$AB_SNDBUFFER, R2	:
5E		04 C2 0001E	SUBL2 #4, SP	:
00000000V	00	00 FB 00021	CALLS #0, NML_OPEN_CONFIG_LINK	: 0326
	00000000G	00 9F 00028	PUSHAB NML\$GL_RCVDATLEN	: 0331
	00000000G	00 9F 0002E	PUSHAB NML\$AB_RCVBUFFER	:
		30 DD 00034	PUSHL #48	:
7E		63 3C 00036	MOVZWL NML\$W_CONFIG_CHAN, -(SP)	:
64		04 FB 00039	CALLS #4, NML_CONFIG_QIO	:
6E	0200	8F 3C 0003C	MOVZWL #512, MSG_LEN	: 0339
	4004	8F BB 00041	PUSHR #^M<R2,SP>	: 0340
		31 DD 00045	PUSHL #49	:
7E		63 3C 00047	MOVZWL NML\$W_CONFIG_CHAN, -(SP)	:
64		04 FB 0004A	CALLS #4, NML_CONFIG_QIO	:
02		62 91 0004D	CMPB NML\$AB_SNDBUFFER, #2	: 0349
		20 12 00050	BNEQ 2\$:

NML\$FORWARD
V04-000

Network Management Listener module to forward N 16--Sep-1984 00:15:46
NML\$CALL_NI_CONFIG Routine to talk to NI Conf 14-Sep-1984 12:50:09

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLFORWRD.B32;1

Page 13
(4)

6E	0200	8F	3C	00052	1\$:	MOVZWL	#512, MSG_LEN	:	0353
	4004	8F	BB	00057		PUSHR	#^M<R2,SP>	:	0354
		31	DD	0005B		PUSHL	#49	:	
7E		63	3C	0005D		MOVZWL	NML\$W_CONFIG_CHAN, -(SP)	:	
64		04	FB	00060		CALLS	#4, NML_CONFIG_QIO	:	
80	8F	62	91	00063		CMPB	NML\$AB_SNDBUFFER, #128	:	0362
		10	13	00067		BEQL	3\$:	
		6E	DD	00069		PUSHL	MSG_LEN	:	0368
		52	DD	0006B		PUSHL	R2	:	
65		02	FB	0006D		CALLS	#2, NML\$SEND	:	
		E0	11	00070		BRB	1\$:	0351
		6E	DD	00072	2\$:	PUSHL	MSG_LEN	:	0375
		52	DD	00074		PUSHL	R2	:	
65		02	FB	00076		CALLS	#2, NML\$SEND	:	
		04	00079	3\$:		RET		:	0376

; Routine Size: 122 bytes, Routine Base: \$CODE\$ + 015B


```
381 0377 1 %SBTTL 'nml_open_config_link Open link to NICONFIG'
382 0378 1 ROUTINE nml_open_config_link: NOVALUE =
383 0379 1
384 0380 1 !++
385 0381 1 FUNCTIONAL DESCRIPTION:
386 0382 1 This routine opens a logical link to the NI Configurator Module.
387 0383 1
388 0384 1 FORMAL PARAMETERS:
389 0385 1 NONE
390 0386 1
391 0387 1 IMPLICIT INPUTS:
392 0388 1
393 0389 1 IMPLICIT OUTPUTS:
394 0390 1 nml$w_config_chan = Channel to Configurator Module.
395 0391 1 nml$w_mbxchan = Channel to mailbox.
396 0392 1
397 0393 1 ROUTINE VALUE:
398 0394 1 COMPLETION CODES:
399 0395 1 NONE
400 0396 1
401 0397 1 SIDE EFFECTS:
402 0398 1 NONE
403 0399 1
404 0400 1 --
405 0401 1
406 0402 2 BEGIN
407 0403 2
408 0404 2 LOCAL
409 0405 2 iosb : iosb, ! IO status block
410 0406 2 status;
411 0407 2
412 0408 2 !! OWN
413 0409 2 OBJNAM_DESC : BBLOCK [DSC$C_S_BLN]
414 0410 2 INITIAL (%CHARCOUNT-('$NICONFIG'),
415 0411 2 UPLIT PSECT ($OWNS) (%ASCII '$NICONFIG'));
416 0412 2
417 0413 2
418 0414 2 ! If there is already a link to the NI Configurator Module, just return.
419 0415 2
420 0416 2 IF .nml$w_config_chan NEQ 0 THEN
421 0417 2 RETURN ss$normal;
422 0418 2
423 0419 2 status = LIB$ASN_WTH_MBX ( %ASCII '_NET:',
424 0420 2 0,0,
425 0421 2 nml$w_config_chan,
426 0422 2 nml$w_mbxchan);
427 0423 2
428 0424 2 nml_chkerr (.status, 0);
429 0425 2
430 0426 2 status = $QIOW (
431 0427 2 FUNC = io$access,
432 0428 2 CHAN = .nml$w_config_chan,
433 0429 2 IOSB = iosb,
434 0430 2 P2 = nml$q_ncb);
435 0431 2
436 0432 2 nml_chkerr (.status, iosb);
437 0433 2
```

! Assign channel to NETACP
! mailbox MAXMSG, BUFQUO (ignored)
! Channel to Configurator Module
! Channel to mailbox

! Check completion status and
! signal if there's an error

! Request connect
! Use assigned channel

! Network connect block

! Check completion status and
! signal if error.


```

: 438 P 0434 2 status = $QIOW (
: 439 P 0435 2 FUNC = io$_readvblk,      ! Request read on mailbox
: 440 P 0436 2 CHAN = .nml$w_mbxchan,    ! Use assigned channel
: 441 P 0437 2 IOSB = iosb,
: 442 P 0438 2 P1 = nml$a_mbxmsg,        ! Buffer to contain mailbox message
: 443 0439 2 P2 = nml$c_maxmbxmsg);      ! Size maximum on mailbox message
: 444 0440 2
: 445 0441 2 nml_chkerr (.status, iosb);  ! Check completion status and
: 446 0442 2                                     ! signal if error.
: 447 0443 2 IF .nml$a_mbxmsg [0] NEQ msg$_confirm THEN
: 448 0444 2     nml_chkerr (ss$_endoffile, 0); ! The connect was not accepted.
: 449 0445 2
: 450 0446 2 RETURN;
: 451 0447 1 END;                        ! of nml_open_config_link
```

```

                                .PSECT $SPLITS,NOWRT,NOEXE,2
00 00 00 3A 54 45 4E 5F 00090 P.AAP: .ASCII \_NET:\<0><0><0>
                                010E0005 00098 P.AAO: .LONG 17694725
                                00000000' 0009C .ADDRESS P.AAP
```

```
.EXTRN SYSS$QIOW
```

```
.PSECT $CODE$,NOWRT,2
```

```
003C 00000 NML_OPEN_CONFIG_LINK:
```

```

55 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5      : 0378
54 00000000V 00 9E 00009 MOVAB SYSS$QIOW, R5
53 00000000' 00 9E 00010 MOVAB NML_CHKERR, R4
5E          08 C2 00017 MOVAB NML$W_CONFIG_CHAN, R3
          63 B5 0001A SUBL2 #8, SP
          78 12 0001C TSTW NML$W_CONFIG_CHAN      : 0416
          02 A3 9F 0001E BNEQ 1$
          53 DD 00021 PUSHAB NML$W_MBXCHAN          : 0419
          7E 7C 00023 PUSHL R3
          00 9F 00025 CLRQ -(SP)
00000000G 00 05 FB 0002B PUSHAB P.AAO
52          50 D0 00032 CALLS #5, LIB$ASN_WTH_MBX
          7E D4 00035 MOVL R0, STATUS
          52 DD 00037 CLRL -(SP)
          64 02 FB 00039 PUSHL STATUS
          7E 7C 0003C CALLS #2, NML_CHKERR          : 0424
          7E 7C 0003E CLRL -(SP)
          00 9F 00040 CLRL -(SP)
          7E 7C 00046 PUSHAB NML$Q_NCB
          7E D4 00048 CLRL -(SP)
          20 AE 9F 0004A PUSHAB IOSB
          32 DD 0004D PUSHL #50
          7E 63 3C 0004F MOVZWL NML$W_CONFIG_CHAN, -(SP)
          7E D4 00052 CLRL -(SP)
          65 0C FB 00054 CALLS #12, SYSS$QIOW
          52 50 D0 00057 MOVL R0, STATUS
          64 4004 8F BB 0005A PUSHR #^M<R2,SP>
          02 FB 0005E CALLS #2, NML_CHKERR          : 0432
```

```
Network Management Listener module to forward N
nml_open_config_link Open link to NICONFIG
```

16-Sep-1984 00:15:46
14-Sep-1984 12:50:09

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLFORWRD.B32;1

Page 16
(5)

		7E	7C	00061	CLRQ	-(SP)
		7E	7C	00063	CLRQ	-(SP)
7E	C8	8F	9A	00065	MOVZBL	#200, -(SP)
	0E	A3	9F	00069	PUSHAB	NML\$A_MBXMSG
		7E	7C	0006C	CLRQ	-(SP)
	20	AE	9F	0006E	PUSHAB	IOSB
		31	DD	00071	PUSHL	#49
7E	02	A3	3C	00073	MOVZWL	NML\$W_MBXCHAN, -(SP)
		7E	D4	00077	CLRL	-(SP)
65		0C	FB	00079	CALLS	#12, SYSSQIOW
52		50	DO	0007C	MOVL	R0, STATUS
	4004	8F	BB	0007F	PUSHR	#^M<R2,SP>
64		02	FB	00083	CALLS	#2, NML_CHKERR
31	0E	A3	91	00086	CMPB	NML\$A_MBXMSG, #49
		0A	13	0008A	BEQL	1\$
		7E	D4	0008C	CLRL	-(SP)
7E	0870	8F	3C	0008E	MOVZWL	#2160, -(SP)
64		02	FB	00093	CALLS	#2, NML_CHKERR
		04	00096	1\$:	RET	

; Routine Size: 151 bytes, Routine Base: \$CODE\$ + 01D5


```

: 453 0448 1 %SBTTL 'nml_config_qio Issue QIO to NICONFIG'
: 454 0449 1 ROUTINE nml_config_qio (forward_chan, function, buffer_addr, buffer_len) =
: 455 0450 1
: 456 0451 1 ++
: 457 0452 1 FUNCTIONAL DESCRIPTION:
: 458 0453 1 Issue a read or a write on the logical link to NICONFIG.
: 459 0454 1
: 460 0455 1 FORMAL PARAMETERS:
: 461 0456 1 forward_chan - channel on which to do QIO
: 462 0457 1 function - io$_readvblk or io$_writevblk
: 463 0458 1 buffer_addr - Address of buffer from which to put or get data.
: 464 0459 1 buffer_len - byte count of data to write, or size of buffer
: 465 0460 1 to receive data.
: 466 0461 1
: 467 0462 1 OUTPUTS:
: 468 0463 1 buffer_len - length of data read (if it's a read).
: 469 0464 1
: 470 0465 1 IMPLICIT INPUTS:
: 471 0466 1
: 472 0467 1
: 473 0468 1 IMPLICIT OUTPUTS:
: 474 0469 1
: 475 0470 1
: 476 0471 1 ROUTINE VALUE:
: 477 0472 1 COMPLETION CODES:
: 478 0473 1
: 479 0474 1 NONE
: 480 0475 1
: 481 0476 1 SIDE EFFECTS:
: 482 0477 1
: 483 0478 1 NONE
: 484 0479 1
: 485 0480 1 --
: 486 0481 1
: 487 0482 2 BEGIN
: 488 0483 2
: 489 0484 2 LOCAL
: 490 0485 2 status,
: 491 0486 2 iosb: $iosb;
: 492 0487 2
: 493 0488 2 IF .function EQL io$_writevblk THEN
: 494 0489 2 nml$debug_msg (dbg$c_netio,
: 495 0490 2 .buffer_addr,
: 496 0491 2 .buffer_len,
: 497 0492 2 %ASCII 'NICE message forwarded to NICONFIG or MOM');
: 498 P 0493 2 status = $QIOW (CHAN = .forward_chan,
: 499 P 0494 2 FUNC = .function,
: 500 P 0495 2 IOSB = iosb,
: 501 P 0496 2 P1 = .buffer_addr,
: 502 0497 2 P2 = ..buffer_len);
: 503 0498 2
: 504 0499 2 nml_chkerr (.status, iosb); ! Check completion status and
: 505 0500 2 ! signal if error.
: 506 0501 2 IF .function EQL io$_readvblk OR
: 507 0502 2 .function EQL (io$_readvblk OR io$m_now) THEN
: 508 0503 2 BEGIN
: 509 0504 2 .buffer_len = .iosb [ios$w_count];
```

```
: 510      0505      3      nml$debug_msg (dbg$netio,  
: 511      0506      3      .buffer_addr,  
: 512      0507      3      .buffer_len,  
: 513      0508      3      %ASCII 'NICE' message received from NICONFIG or MOM');  
: 514      0509      2      END;  
: 515      0510      2      RETURN nml$_sts_suc;  
: 516      0511      2  
: 517      0512      1      END;                                ! of nml_config_qio
```

```
6F 66 20 65 67 61 73 73 65 6D 20 45 43 49 4E 000A0 P.AAR: .PSECT $SPLITS,NOWRT,NOEXE,2  
4F 43 49 4E 20 6F 74 20 64 65 64 72 61 77 72 000AF .ASCII \NICE message forwarded to NICONFIG or MO\  
4F 4D 20 72 6F 20 47 49 46 4E 000BE  
00 00 00 4D 000C8  
010E0029 000CC P.AAQ: .ASCII \M\<0><0><0>  
00000000' 000D0 .LONG 17694761  
00000000' 000D0 .ADDRESS P.AAR  
65 72 20 65 67 61 73 73 65 6D 20 45 43 49 4E 000D4 P.AAT: .ASCII \NICE message received from NICONFIG or M\  
43 49 4E 20 6D 6F 72 66 20 64 65 76 69 65 63 000E3  
4D 20 72 6F 20 47 49 46 4E 4F 000F2  
00 00 4D 4F 000FC  
010E002A 00100 P.AAS: .ASCII \OM\<0><0>  
00000000' 00104 .LONG 17694762  
00000000' 00104 .ADDRESS P.AAT
```

```
.PSECT $CODE$,NOWRT,2  
0004 00000 NML_CONFIG_Q10:  
52 00000000G 00 9E 00002 .WORD Save R2 : 0449  
5E 08 C2 00009 MOVAB NML$DEBUG_MSG, R2  
30 08 AC D1 0000C SUBL2 #8, SP : 0488  
11 12 00010 CMPL FUNCTION, #48  
00000000' 00 9F 00012 BNEQ 1$ : 0491  
10 BC DD 00018 PUSHAB P.AAQ  
0C AC DD 0001B PUSHL @BUFFER_LEN : 0490  
7E D4 0001E PUSHL BUFFER_ADDR : 0489  
62 04 FB 00020 CLRL -(SP)  
7E 7C 00023 1$: CALLS #4, NML$DEBUG_MSG : 0497  
7E 7C 00025 CLRL -(SP)  
10 BC DD 00027 CLRL -(SP)  
0C AC DD 0002A PUSHL @BUFFER_LEN  
7E 7C 0002D PUSHL BUFFER_ADDR  
20 AE 9F 0002F CLRL -(SP)  
7E 04 AC 7D 00032 PUSHAB IOSB  
00000000G 00 0C FB 00038 MOVQ FORWARD_CHAN, -(SP)  
4001 8F BB 0003F CLRL -(SP)  
00000000V 00 02 FB 00043 CALLS #12, SYSSQIOW : 0499  
31 08 AC D1 0004A PUSHR #^M<R0,SP>  
0A 13 0004E CALLS #2, NML_CHKERR : 0501  
00000071 8F 08 AC D1 00050 CMPL FUNCTION, #49  
16 12 00058 BEQL 2$ : 0502  
10 BC 02 AE 3C 0005A 2$: CMPL FUNCTION, #113  
MOVZWL IOSB+2, @BUFFER_LEN : 0504
```


Network Management Listener module to forward N
nml_config_qio Issue QIO to NICONFIG

H 14
16-Sep-1984 00:15:46
14-Sep-1984 12:50:09

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLFORWRD.B32;1

Page 19
(6)

```

00000000' 00 9F 0005F PUSHAB P.AAS
          10 BC DD 00065 PUSHL @BUFFER_LEN
          OC AC DD 00068 PUSHL BUFFER_ADDR
          7E D4 0006B CLRL -(SP)
62 04 04 FB 0006D CALLS #4, NML$DEBUG_MSG
50 01 D0 00070 3$: MOVL #1, R0
          04 00073 RET

```

: 0507
:
: 0506
:
: 0505
:
:
: 0510
:
: 0512

; Routine Size: 116 bytes, Routine Base: \$CODE\$ + 026C

NML
VO4

```
519 0513 1 %SBTTL 'nml_chkerr      Check QIO completion status'
520 0514 1 ROUTINE nml_chkerr (status, iosb): NOVALUE =
521 0515 1
522 0516 1 ++
523 0517 1 FUNCTIONAL DESCRIPTION:
524 0518 1     This routine is called to check the status returns for QIOs
525 0519 1     on the logical link to NICONFIG.  If there is an error, a response
526 0520 1     message is built, and sent to NCP via the handler.
527 0521 1
528 0522 1 FORMAL PARAMETERS:
529 0523 1     status - the completion status of the QIO
530 0524 1     iosb - the address of the iosb for the QIO.
531 0525 1
532 0526 1 ROUTINE VALUE:
533 0527 1 COMPLETION CODES:
534 0528 1
535 0529 1     NONE
536 0530 1
537 0531 1 --
538 0532 1
539 0533 2 BEGIN
540 0534 2
541 0535 2 MAP
542 0536 2     iosb:      REF $iosb;
543 0537 2
544 0538 2 LOCAL
545 0539 2     msgsize;
546 0540 2
547 0541 2 IF .status AND
548 0542 2     .iosb NEQ 0 THEN
549 0543 2     status = .iosb [ios$w_status];
550 0544 2 IF NOT .status THEN
551 0545 3 BEGIN
552 0546 3
553 0547 3     Get rid of the logical link to NICONFIG, and clear the channel number
554 0548 3     so the next request to NICONFIG causes NML to establish another logical
555 0549 3     link to NICONFIG.
556 0550 3
557 0551 3     $DASSGN (CHAN = .nml$w_config_chan);
558 0552 3     nml$w_config_chan = 0;
559 0553 3
560 0554 3     Send an error response to NCP.
561 0555 3
562 0556 3     nml$ab_msgblock [msb$b_code] = nma$c_sts_ope;
563 0557 3 IF .status EQL ss$_endoffile THEN
564 0558 4 BEGIN
565 0559 4     nml$ab_msgblock [msb$l_flags] = msb$m_msg_fld;
566 0560 4     nml$ab_msgblock [msb$l_text] = nml$_opabterm;
567 0561 4 END
568 0562 3 ELSE
569 0563 4 BEGIN
570 0564 4     nml$ab_msgblock [msb$l_flags] = msb$m_msg_fld OR msb$m_sysm_fld;
571 0565 4     nml$ab_msgblock [msb$l_text] = .status;
572 0566 3 END;
573 0567 3 nml$bld_reply (nml$ab_msgblock, msgsize);
574 0568 3 $signal_msg (nml$ab_sndbuffer, .msgsize);
575 0569 2 END;
```


: 576
: 5770570 2
0571 1 END; ! of nml_chkerr

.EXTRN SYSSDASSGN

			000C	00000	NML_CHKERR:			
	55	00000000'	00	9E	00002	.WORD	Save R2,R3	: 0514
	52	00000000G	00	9E	00009	MOVAB	NML\$W_CONFIG_CHAN, R3	:
	5E		04	C2	00010	MOVAB	NML\$AB_MSGBLOCK, R2	:
	0E		04	E9	00013	SUBL2	#4, SP	:
		04	AC	D5	00017	BLBC	STATUS, 2\$: 0541
		08	AC	D5	00017	TSTL	IOSB	: 0542
			05	13	0001A	BEQL	1\$:
04	AC	08	BC	3C	0001C	MOVZWL	@IOSB, STATUS	: 0543
	50	04	AC	E8	00021	BLBS	STATUS, 5\$: 0544
	7E		63	3C	00025	MOVZWL	NML\$W_CONFIG_CHAN, -(SP)	: 0551
00000000G	00		01	FB	00028	CALLS	#1, SYSSDASSGN	:
			63	B4	0002F	CLR	NML\$W_CONFIG_CHAN	: 0552
04	A2		19	8E	00031	MNEGB	#25, NML\$AB_MSGBLOCK+4	: 0556
00000870	8F	04	AC	D1	00035	CMPL	STATUS, #2160	: 0557
			0D	12	0003D	BNEQ	3\$:
	62		04	D0	0003F	MOVL	#4, NML\$AB_MSGBLOCK	: 0559
0C	A2	00000000G	8F	D0	00042	MOVL	#NML\$OPABTERM, NML\$AB_MSGBLOCK+12	: 0560
			09	11	0004A	BRB	4\$: 0557
	62	44	8F	9A	0004C	MOVZBL	#68, NML\$AB_MSGBLOCK	: 0564
0C	A2	04	AC	D0	00050	MOVL	STATUS, NML\$AB_MSGBLOCK+12	: 0565
		4004	8F	BB	00055	PUSHR	#*M<R2,SP>	: 0567
00000000G	00		02	FB	00059	CALLS	#2, NML\$BLD_REPLY	:
			6E	DD	00060	PUSHL	MSGSIZE	: 0568
		00000000G	00	9F	00062	PUSHAB	NML\$AB_SNDBUFFER	:
		01F90000	8F	DD	00068	PUSHL	#33095680	:
00000000G	00		03	FB	0006E	CALLS	#3, LIB\$SIGNAL	:
			04	00075	5\$:	RET		: 0571

; Routine Size: 118 bytes, Routine Base: \$CODE\$ + 02E0

: 578
: 579
: 580
: 581
0572 1
0573 1
0574 1 END
0575 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	304	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$PLITS	264	NOVEC, NGWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	854	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics					
File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
-\$255\$DUA28:[NML.OBJ]NMLLIB.L32;1	341	31	9	27	00:00.1
-\$255\$DUA28:[SHRLIB]NMALIBRY.L32;1	887	3	0	47	00:00.2
-\$255\$DUA28:[SHRLIB]NET.L32;1	1279	0	0	63	00:00.3
-\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	16	0	581	00:03.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:NMLFORWRD/OBJ=OBJ\$:NMLFORWRD MSRC\$:NMLFORWRD/UPDATE=(ENH\$:NMLFORWRD)

Size: 854 code + 568 data bytes

Run Time: 00:18.4

Elapsed Time: 00:58.4

Lines/CPU Min: 1877

Lexemes/CPU-Min: 14193

Memory Used: 135 pages

Compilation Complete

0283 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

